TITLE:

基底変換によるb-Grobner基底の計算について (グレブナ-基底の理論的有効性と実践的有効性)

AUTHOR(S):

伊藤, 雅史

CITATION:

伊藤, 雅史. 基底変換によるb-Grobner基底の計算について (グレブナ-基底の理論的有効性と実践的有効性). 数理解析研究所講究録 2002, 1289: 81-93

ISSUE DATE:

2002-09

URL:

http://hdl.handle.net/2433/42507

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録 1289 巻 2002 年 81-93

基底変換による b–Gröbner 基底の計算について

伊藤 雅史 (Masafumi Ito)
東京理科大学 工学研究科
Department of Engineering, Tokyo University of Science

要旨

整数計画問題に対する Gröbner 基底を用いたアプローチは, 1991 年の P. Conti と C. Traverso のアルゴリズム (C-T アルゴリズム) に端を発する. このアルゴリズムは, トーリックイデアルというある特殊なイデアルとその Gröbner 基底の性質を利用している. アルゴリズムの内部において Gröbner 基底の計算が必要なために十分に実用的なサイズの問題を解くことはできないものの, 整数計画問題, あるいは線形計画問題を代数的に分析する強力なツールとして, 近年盛んに研究されている. 特に, $0-1$ 整数計画問題として定式化された様々な組合せ最適化問題の構造の解明や計算量の解析などで, いくつかの興味深い結果を見ることができる. 今後, さらに多くの問題に対してこのような研究がなされていくと考えられるが, やはり Gröbner 基底計算の困難さは 1 つの障害になっている. 組合せ最適化問題の中には, 小さな例であってもそれを整数計画問題として定式化すると問題のサイズが非常に大きくなるものも少なくない. 例えば, 巡回セールスマン問題 (TSP) などでは, グラフの点数が 5 点の問題に対してすでに Gröbner 基底の計算が計算できないのが現状である.

トーリックイデアルに特化した Gröbner 基底の計算に関する研究はいくつかなされているが, さらに議論を整数計画問題に限定すると, 1997 年の R. Weismantel と R. Thomas の結果が大変興味深い. 彼らは C-T アルゴリズムにおいては必ずしも Gröbner 基底のすべての元が必要とされるわけではないことに注目し, トーリックイデアルにある次数付けを与えることにより, C-T アルゴリズムで必要とされる元を特徴づけることに成功した. なお, C-T アルゴリズムで必要とされる Gröbner 基底の部分集合は b–Gröbner 基底と呼ばれる. 本論文では, この b–Gröbner 基底の計算の高速化について考察する. Thomas と Weismantel も b–Gröbner 基底計算のアルゴリズムを提案したが, Buchberger のアルゴリズムを元にしており, それゆえに莫大な計算時間を要することがわかっている. そこで本研究では FGLM アルゴリズムおよび Gröbner Walk アルゴリズムという, 基底変換のテクニックを用いている.

Keywords 整数計画問題, Conti-Traverso アルゴリズム, b–Gröbner 基底, 基底変換アルゴリズム

1 Conti-Traverso (C-T) アルゴリズム

本論文では, 以下のような整数計画問題を考える.

$$\mathrm{IP}_{A,c}(b) = \min\{c \cdot x \ : \ Ax = b, x \in \mathbb{N}^n\}.$$

ただし, $A = [a_{ij}] \in \mathbb{N}^{d\times n}$, $b \in \mathbb{N}^d$, $c \in \mathbb{R}^n_+$ とする. また, $a_1, \ldots, a_n$ を A の列ベクトルとする. 一般に, A は係数行列, b は右辺ベクトル, そして c はコストベクトルと呼ばれる. いま $u \in \mathbb{N}^n$ が

$Au = b$ を満たすとき, u を実行可能解という. $\mathrm{IP}_{A,c}(b)$ に実行可能解が存在しないとき, この問題は実行不能であるという. 明らかに $\mathrm{IP}_{A,c}(b)$ が実行可能であるための必要十分条件は, b がモノイド $\mathcal{M}(A) := \{Au : u \in \mathbb{N}^n\}$ に属することである. 本論文では, 前提として $\{x \geq 0 \ : \ Ax = 0\} = \{0\}$ が成り立つものとする. これは実行可能解の集合 $\{u \in \mathbb{N}^n : Au = b\}$ が有限集合であることと同値である.

まず, C-T アルゴリズムについて説明する. k を任意の体とし, 多項式環 $k[X] := k[X_1, \ldots, X_n]$, $k[Y] := k[Y_1, \ldots, Y_d]$, および $k[X, Y] := k[X_1, \ldots, X_n, Y_1, \ldots, Y_d]$ を考える. 準同型

$$\phi : k[X] \to k[Y], \quad X_i \mapsto Y^{a_i} := Y_1^{a_{1i}} \ldots Y_d^{a_{di}}$$

の核を A のトーリックイデアルとよび, I_A で表す. すなわち,

$$I_A := \{f \in k[X] : \phi(f) = 0\}.$$

I_A は 2 項式 (項と項の差で表される多項式) で生成されるイデアル (2 項式イデアル) であることが知られている.

いま, $\succ$ を $k[X]$ の任意の項順序とし, $\succ_c$ を次のように定める. すなわち任意の X^α, X^β に対して

$$X^\alpha \succ_c X^\beta \ \Leftrightarrow \begin{cases} \alpha \cdot c > \beta \cdot c \\ \text{もしくは} \\ \alpha \cdot c = \beta \cdot c \text{ かつ } X^\alpha \succ X^\beta. \end{cases}$$

この項順序を, コストベクトル c を細分化した順序と呼ぶ. C-T アルゴリズムのアイデアは単純明快である.

C-T アルゴリズム (condenced version)
STEP 1: $\mathrm{IP}_{A,c}(b)$ の実行可能解を求めて u とおく.
STEP 2: I_A の $\succ_c$ に関する Gröbner 基底 $\mathcal{G}$ を求める.
STEP 3: 項 X^u を $\mathcal{G}$ で割り, 余りを X^v とおく. このとき v が $\mathrm{IP}_{A,c}(b)$ の最適解. □

I_A は 2 項式イデアルなのでその Gröbner 基底も 2 項式の集合としてとれること, よって項を Gröbner 基底で割った余りもまた項であることに注意する. 上に述べた C-T アルゴリズム (condenced version) の難点は次の 2 点である.

(P1) $\mathrm{IP}_{A,c}(b)$ の実行可能解を求めることは, 一般に困難である.

(P2) I_A の Gröbner 基底の計算に不可欠な生成元が明示的でない.

実は I_A の代数的な性質によりこれらの難点がうまく解決される, という点も, C-T アルゴリズムの鮮やかさの 1 つである. それぞれについて解説する.
(P1). $\mathrm{IP}_{A,c}(b)$ に対して, よりサイズの大きな整数計画問題 $\mathrm{IP}_{\tilde{A},\tilde{c}}(b)$ を考える. ここで,

$$\tilde{A} := [A, I] \ \ I \text{ は } d \times d \text{ 単位行列}$$
$$\tilde{c} := (c, M, \ldots, M) \in \mathbb{R}^{n+d} \ \ M \text{ は十分に大きな実数}$$

である. 問題 $\mathrm{IP}_{\tilde{A},\tilde{c}}(b)$ の特徴として次が知られている.

- 自明な実行可能解 $(0,\ldots,0,b_1,\ldots,b_d)$ をもつこと.
- $\mathrm{IP}_{\tilde{A},\tilde{c}}(b)$ の最適解 $(v_1,\ldots,v_n,v_{n+1},\ldots,v_{n+d})$ において, $v_{n+1}=\cdots=v_{n+d}=0$ ならば $(v_1,\ldots,v_n)$ は $\mathrm{IP}_{A,c}(b)$ の最適解. そうでないならば $\mathrm{IP}_{A,c}(b)$ は実行不能.

これらの性質により, $\mathrm{IP}_{\tilde{A},\tilde{c}}(b)$ を考えることにより, 変数の数は多くなるものの, 難点 (P1) は克服される.
(P2). C-T アルゴリズムでは, イデアル I_A の Gröbner 基底を求めなければならない. この方法としてはいくつか知られているが, ここでは Sturmfels[11] で述べられている方法について説明する. $\mathcal{T}_X$, $\mathcal{T}_Y$, $\mathcal{T}_{X,Y}$ をそれぞれ $k[X]$, $k[Y]$, $k[X,Y]$ の項の集合とする. $k[X,Y]$ 上の項順序 $\succ$ が $Y>X$ であるような消去項順序であるとは, 任意の $m_1\in\mathcal{T}_{X,Y}\backslash\mathcal{T}_X$ および $m_2\in\mathcal{T}_X$ に対して $m_1\succ m_2$ が成り立つことをいう. 消去定理 [5] により, I_A の Gröbner 基底は次のようにして求められることが知られている.

Note 1.1. *[11]* $\succ$ を $Y>X$ であるような消去項順序とする. $\mathcal{F}$ をイデアル

$$J:=(X_1-Y^{a_1},\ldots,X_n-Y^{a_n})$$

の $\succ$ に関する *Gröbner* 基底とする. このとき, $\mathcal{F}\cap k[X]$ は I_A の *Gröbner* 基底である.

$\succ$ を Note 1.1 のような消去項順序で, かつ $k[X]$ に制限したときに $\succ_c$ と一致するように定めることにより, C-T アルゴリズムで求めるべき Gröbner 基底を計算することができる.
以上により, 実行可能解が自明でないような整数計画問題については, 拡大された係数行列 $\tilde{A}$ のトーリックイデアルに対して, (P2) で述べた方法によりその Gröbner 基底を計算すればよい. ところが, 次の事実こそ C-T アルゴリズムの巧妙さの 1 つである.

Note 1.2. *[11]* イデアル J は $\tilde{A}$ のトーリックイデアルである.

以上を考慮した C-T アルゴリズム (original version) は次のようになる. ここで $\succ$ は Note 1.1 のような消去項順序で, かつ $k[X]$ に制限したときに $\succ_c$ と一致するようなものとする. この定め方がコストベクトル $\tilde{c}$ の定め方と矛盾しないことに注意する.

C-T アルゴリズム (original version)
STEP 1: $I_{\tilde{A}}$ の $\succ$ に関する Gröbner 基底 $\mathcal{F}$ を求める.
STEP 2: 項 Y^b を $\mathcal{F}$ で割った余りを X^vY^u とする. $u=0$ であるならば v は $\mathrm{IP}_{A,c}(b)$ の最適解. $u\neq 0$ ならば $\mathrm{IP}_{A,c}(b)$ は実行不能. □

このアルゴリズムは RISA/ASIR などの代数計算システム上で容易に実装できるが, その計算時間のほとんどが Gröbner 基底の計算に費やされる. 係数行列が $0-1$ 行列の場合, 8×28 程度の問題までは解くことができるが, 一般の行列の場合は, 4×6 程度の問題でも解くことができない場合がある. これは Gröbner 基底の計算が, 多項式の次数も大きく依存していることを意味している.

2 b–Gröbner 基底

トーリックイデアルに特化した Gröbner 基底計算についてもいくつかの研究がなされているが, R. Thomas と R. Weismantel は整数計画問題に対して b–Gröbner 基底を提案した [13]. 彼らは C-T

アルゴリズム (condenced version) において実行可能解に対応する項 X^u を Gröbner 基底 $\mathcal{G}$ で割る際, 必ずしも $\mathcal{G}$ のすべての元が必要とされているわけではないことを指摘した. そして, $\mathcal{G}$ の元の中で不必要なものを特徴づけることに成功した.

1975 年, Graver は整数計画問題 $\mathrm{IP}_{A,c}(b)$ に対する test set という概念を提案した. これは次のように定義される.

Definition 2.1. $T \subset \mathbb{Z}^n \backslash \{0\}$ が $\mathrm{IP}_{A,c}(b)$ の *test set* であるとは, 次を満たすことである:

- $u \in \mathbb{N}^n$ が実行可能解でありかつ最適解でないならば, $v \in T$ が存在して $u - v$ は実行可能解, かつ $c \cdot u > c \cdot (u - v)$,
- $u \in \mathbb{N}^n$ が最適解ならば, すべての $v \in T$ に対して $u - v$ は実行可能解でない.

C-T アルゴリズムの正当性から, I_A の Gröbner 基底が $\mathrm{IP}_{A,c}(b)$ の test set に対応することは明らかであるが, より広く, 次がいえる.

Note 2.1. I_A の *Gröbner* 基底 $\mathcal{G}$ に対して, 集合 $\{u - v : X^u - X^v \in \mathcal{G}\}$ は

$$\mathrm{IP}_{A,c} := \{\mathrm{IP}_{A,c}(b) : b \in \mathcal{M}(A)\}$$

のすべての問題に対する *test set* である.

C-T アルゴリズムにおいては Gröbner 基底でなくとも, test set に対応する 2 項式の集合でありさえすればよいということがわかる. そこで, $\mathcal{G}$ の中から $\mathrm{IP}_{A,c}(b)$ の test set に対応する元を抽出する Thomas らの方法について述べる.

トーリックイデアルに次のような次数付けを導入する. すなわち, $k[X]$ の項に対してその A-次数を

$$\deg_A(X^u) := Au \in \mathbb{Z}^d$$

で定める. 多項式 $f \in k[X]$ に対して, そのすべての項の A-次数がすべて等しいときに f は斉次であるといい, そのとき f の A-次数 $\deg_A(f)$ を任意の項の A-次数とする. トーリックイデアル I_A においては, 2 項式はすべて斉次であることに注意する. さらに, $\mathbb{Z}^d$ 上の自然な半順序 $\leq_{\mathcal{M}}$ を

$$u \leq_{\mathcal{M}} v \Leftrightarrow v - u \in \mathcal{M}(A)$$

で定める.

A-次数に関する重要な性質をいくつか述べておく.

Note 2.2. *[13]* A-次数に関して以下が成り立つ:

(i) X^u が X^v を割り切るならば $\deg_A(X^u) \leq_{\mathcal{M}} \deg_A(X^v)$.

(ii) 斉次で零でない多項式 $f, g \in I_A$ が $f + g \neq 0$, $\deg_A(f) = \deg_A(g)$ を満たすならば, $\deg_A(f + g) = \deg_A(f)$.

(iii) 斉次で零でない多項式 $f, g \in I_A$ に対して $\deg_A(fg) = \deg_A(f) + \deg_A(g)$.

(iv) 斉次で零でない多項式 $f, p \in I_A$ について, g が f を p で割った余りであるならば, $\deg_A(f) = \deg_A(g)$ であり, かつ $\deg_A(f) \geq_{\mathcal{M}} \deg_A(p)$.

(v) 斉次な多項式 $f,g \in I_A$ が $\deg_A(f) \not\leq_{\mathcal{M}} b$ を満たすならば $\deg_A(\mathrm{Spol}(f,g)) \not\leq_{\mathcal{M}} b$. ただし $\mathrm{Spol}(f,s)$ は f と g の S 多項式 *[5]*.

$\{X^u - X^v : Au = Av\}$ がトーリックイデアル I_A の k–ベクトル空間としての基底である [11] ことから, I_A は $\deg_A$ に関して斉次イデアルであることがわかる. さらに, 次の事実は重要である.

Note 2.3. *[13]* $(I_A)_\beta := \{f \in I_A : \deg_A(f) = \beta\}$ と定めると,

$$I_A = \bigoplus_{\beta \in \mathcal{M}(A)} (I_A)_\beta.$$

Thomas らは, C-T アルゴリズムに不要な元を算出しないように以下のように Buchberger のアルゴリズムに手を加えた.

b–Buchberger アルゴリズム

入力: I_A の生成系 F と項順序.
出力: I_A の b–Gröbner 基底.

STEP 1: F のすべてのペア f,g に対して, $\deg_A(\mathrm{Spol}(f,g)) \leq_{\mathcal{M}} b$ であるならば $\mathrm{Spol}(f,g)$ を F で割り, 余りが零でないならば F に加える.
STEP 2: F を $b\mathcal{G}$ として出力. □

このアルゴリズムの出力 $b\mathcal{G}$ を b–Gröbner 基底と呼ぶ. これと同値な定義として次がある.

Definition 2.2. 有限集合 $b\mathcal{G} \subset I_A$ が I_A の $\succ$ に関する *b–Gröbner* 基底であるとは, 任意の $f \in \bigoplus_{\beta \leq_{\mathcal{M}} b} (I_A)_\beta$ に対して, $g \in b\mathcal{G}$ が存在して $\mathrm{HT}_\succ(g)$ が $\mathrm{HT}_\succ(f)$ を割り切る. ただし $\mathrm{HT}_\succ(g)$ は g の $\succ$ に関する先頭項である.

この同値性は Note2.2 から明らかである. また, 次の事実も重要である.

Note 2.4. ある I_A の *Gröbner* 基底 $\mathcal{G}$ に対して

$$b\mathcal{G} = \mathcal{G} \cap \bigoplus_{\beta \leq_{\mathcal{M}} b} I_\beta$$

と表される.

Note 2.5. $b\mathcal{G}$ を I_A の $\succ_c$ に関する *b–Gröbner* 基底とすると, $\{u - v : X^u - X^v \in b\mathcal{G}\}$ は

$$\{\mathrm{IP}_{A,c}(\beta) : \beta \leq_{\mathcal{M}} b\}$$

のすべての問題に対する *test set* である.

ここで b–Buchberger アルゴリズムの難点を 2 点述べておく. 1 点目として, ある $u \in \mathbb{N}^d$ が与えられたときに, $u \leq_{\mathcal{M}} b$ であるかどうかの判定が一般には困難なことである. これは

$$Ax = b - u, \quad x \in \mathbb{N}^n$$

を満たす x を求めるという, 線形 diophantine 方程式と呼ばれる難問である.
2 点目はやはり, Buchberger アルゴリズム同様, 多大な計算時間を要することである.

3 b–Gröbner 基底の計算について

本研究では, b–Gröbner 基底を計算する 2 種類のアルゴリズムを提案する. 我々が注目したのは, 次の事実である.

Note 3.1. 集合 $\{X_i - Y^{a_i} : i = 1, \ldots, n\}$ *(すなわち, イデアル J の生成系)* は任意の $X > Y$ であるような消去項順序に対してすでに *Gröbner* 基底である. 従って *b–Gröbner* 基底としての性質も満たす.

この事実を元に, Thomas らの Buchberger アルゴリズムを元としたアルゴリズムとは異なった発想のアルゴリズムを構築することが, 我々の目的である. ここで用いるのは基底変換というテクニックである. 基底変換アルゴリズムとは, ある項順序に関する Gröbner 基底が与えられたときに, それを他の項順序に関する Gröbner 基底に変換するアルゴリズムのことをいい, よく知られているものとして FGLM アルゴリズム [7] と Gröbner Walk アルゴリズム [3] があげられる. 本研究では, これらを元に, b–FGLM アルゴリズムおよび b–Gröbner Walk アルゴリズムを提案する.

なお, アルゴリズムの説明においては, C-T アルゴリズムの original version における b–Gröbner 基底計算を対象とする. すなわち, トーリックイデアルの生成元および実行可能解が既知であるものとし, かつその生成元が求めるべき項順序とは異なる項順序に関してすでに b–Gröbner 基底であるものとする. ただし, 記号の煩雑さを防ぐために問題を $\mathrm{IP}_{A,c}(b)$ と表し, トーリックイデアルも $I_A \subset k[X]$ で表す.

3.1 b–FGLM アルゴリズム

FGLM アルゴリズム [7] は, 0 次元イデアルに対してのみ用いることができる基底変換アルゴリズムである. なお, イデアル $I \subset k[X]$ が 0 次元であるとは, 次の同値な条件を満たすことである:

- 代数多様体 $\mathrm{V}(I) := \{a \in k^n : f(a) = 0 \ \forall f \in I\}$ の次元が 0 である.
- 剰余環 $k[X]/I$ の k–ベクトル空間として有限次元である.

残念ながら, トーリックイデアルは本研究の仮定の下では 0 次元イデアルではない. 我々はまず, この点を克服する必要がある.

FGLM アルゴリズムの元になっているのは以下の事実である:

- I の $\succ$ に関する先頭項イデアルを $\mathrm{HT}(I) := (\mathrm{HT}(f) : f \in I)$ で定めるとき, $\{t \in \mathcal{T}_X : t \notin \mathrm{HT}(I)\}$ は $k[X]/I$ の k–ベクトル空間としての基底となる.
- G を $\succ$ に関する I の Gröbner 基底とする. また, $f \in k[X]$ に対して $\bar{f}$ を, f を G で割った余りとする. このとき, $\{\bar{f} : f \in k[X]\}$ は $k[X]/I$ の完全代表系である. また, $f \in k[X]$ に対してある $c_1, \ldots, c_r \in k$ および $f_1, \ldots, f_r \in k[X]$ が存在して $\bar{f} = \sum_{i=1}^r c_i \bar{f}_i$ を満たすならば, $f - \sum_{i=1}^r c_i f_i \in I$ が成り立つ.

アルゴリズムは以下の通りである.

FGLM アルゴリズム

入力: $\succ_1, \succ_2$: 項順序, G_1 : $\succ_1$ に関する Gröbner 基底.
出力: G_2 : $\succ_2$ に関する Gröbner 基底.

STEP 0: $T := \mathcal{T}_X$, $B := \emptyset \subset \mathcal{T}_X \times k[X]$, $G := \emptyset$ とする.
STEP 1: $T = \emptyset$ ならば STEP 6 へ. そうでないならば T の $\succ_2$ に関する最小の元を m とおく.
STEP 2: m を項順序 $\succ_1$ を用いて G_1 で割った余りを $\bar{m}$ とおく.
STEP 3: いま $B = \{(f_1, g_1), \ldots, (f_r, g_r)\}$ と表されているものとして, $\bar{m}$ が $g_1, \ldots, g_r$ の k–線形結合で表されるならば STEP 4 へ. そうでないならば STEP 5 へ.
STEP 4: $\bar{m} = \sum_{i=1}^{r} c_i g_i$ と表されているものとして, $m - \sum_{i=1}^{r} c_i f_i$ を G へ加える. T から m および m で割り切れる項を除いて STEP 1 へ.
STEP 5: $(m, \bar{m})$ を B に加える. T から m を除いて STEP 1 へ.
STEP 6: G を G_2 として出力. □

イデアル I が 0 次元イデアルであることにより, アルゴリズムを実行するといずれ T が空集合となり, よってアルゴリズムは終了する.

このアルゴリズムを元に, b–Gröbner 基底を変換するアルゴリズムを構築する.

I_A が 0 次元イデアルではないということの問題点は, STEP 1 から STEP 5 を何度繰り返しても T が空集合にならないことである. そこで, 最初から T を有限集合でとることを考える. ここで次の事実を用いる.

- アルゴリズムの STEP 1 で選ばれた m が $\deg_A(m) \not\leq_{\mathcal{M}} b$ を満たすならば, その G_1 による余りについても $\deg_A(\bar{m}) \not\leq_{\mathcal{M}} b$ がいえる (Note 2.2 から明らか). よって STEP 4 で $m - \sum_{i=1}^{r} c_i f_i$ が G に加えられるならば, その A–次数もまた, 順序 $\leq_{\mathcal{M}}$ に関して b より小さくない.

- 逆に, アルゴリズムの STEP 1 で選ばれた m が $\deg_A(m) \leq_{\mathcal{M}} b$ を満たすならば, STEP 4 で G に加えられる多項式の A–次数も $\leq_{\mathcal{M}}$ に関して b より小さい.

この事実から明らかなように, STEP 0 において

$$T := \{t \in \mathcal{T}_X : \deg_A(t) \leq_{\mathcal{M}} b\}$$

と定めると,

$$G_2 \cap \bigoplus_{\beta \leq_{\mathcal{M}} b} (I_A)_\beta$$

が算出される. これがすなわち我々の求める b–Gröbner 基底である. なお, T を $\{t \in \mathcal{T}_X : \deg_A(t) \leq_{\mathcal{M}} b\}$ を含むような任意の集合と定めたとしても, 得られる G_2 は b–Gröbner 基底の性質を満たすことは明らかである.

FGLM アルゴリズムでトーリックイデアルを扱うことの利点も存在する. トーリックイデアルが 2 項式イデアルであることから, 入力される Gröbner 基底 G_1 が 2 項式の集合でとれることに注目する. 項を 2 項式で割った余りもまた項であることから, STEP 2 で求められる $\bar{m}$ もまた項である. 本来, FGLM アルゴリズムにおいては, STEP 3 の線形結合で表されるかどうかの判定において, 線形方程式系を解く必要がある. しかしながら, いま $g_1, \ldots, g_r$ がすべて項であるということ

は, この判定が $g_1, \ldots, g_r$ のうちで $\bar{m}$ と一致するものが存在するかどうか, という判定のみで済むことを意味する.

b-FGLM アルゴリズムを記述しておく.

b-FGLM アルゴリズム

入力: $\succ_1, \succ_2$: 項順序, $G_1 : \succ_1$ に関する b-Gröbner 基底.
出力: $G_2 : \succ_2$ に関する b-Gröbner 基底.

STEP 0: T を $\{t \in \mathcal{T}_X : \deg_A(t) \leq_{\mathcal{M}} b\}$ を含むような任意の有限集合とし, $B := \emptyset \subset \mathcal{T}_X \times k[X]$, $G := \emptyset$ とする.
STEP 1 – STEP 6: FGLM アルゴリズムに同じ. □

今までの議論から明らかではあるが, 次の定理を示しておく.

Theorem 3.1. *b-FGLM アルゴリズムで出力される G_2 は $\succ_2$ に関する b-Gröbner 基底である.*

証明. まず, アルゴリズムが終了すること, すなわち T が有限集合にとれることを示しておく. 整数計画問題 $\mathrm{IP}_{A,c}(b)$ の実行可能解が有限個しかないことから, A の列ベクトルに 0 は存在しない. ここですべての $i = 1, \ldots, n$ に対して α_j を

$$\lfloor \min_{1 \leq i \leq d} (b_i / a_{ij}) \rfloor$$

と定め,

$$T := \{t_1^{u_1} \ldots t_d^{u_d} y_1^{u_{d+1}} \ldots y_n^{u_{d+n}} \in k[t, y] \ : \ u_i \leq \alpha_i \ \forall 1 \leq i \leq d+n\}.$$

とすると, 明らかに T は STEP 0 の条件を満たす.

G_2 が b-Gröbner 基底であることは, FGLM アルゴリズムの証明と同様に示される. □

b-FGLM アルゴリズムの難点として次の 2 点があげられる:

- $\succ_2$ によっては, STEP 1 において $\succ_2$ に関して最小の項を選ぶことが困難である.
- 問題が大きくなるに従って, T の要素数が爆発的に増大する.

3.2 b-Gröbner Walk アルゴリズム

Gröbner Walk アルゴリズム [3] はトーリックイデアルに対するコストベクトルの幾何的性質を利用したアルゴリズムである. このアルゴリズムを説明するためにはいくつかの定義が必要である.

多項式 $f \in k[X]$ に対して, 指数ベクトルと c との内積が f において最大であるような項の和を f の c に関する initial form と呼び, $\mathrm{in}_c(f)$ で表す. また, イデアル I に対して

$$\mathrm{in}_c(I) := (\mathrm{in}_c(f) : f \in I)$$

と定め, 集合 $F \subset k[X]$ に対して

$$\mathrm{in}_c(F) := \{\mathrm{in}_c(f) : f \in F\}$$

と定める. コストベクトル c と項順序 $\succ$ が I に関して同値であるとは,

$$\mathrm{HT}_{\succ}(I) = \mathrm{in}_c(I)$$

が成り立つことをいい, このとき $\succ \sim c$ と表す. またこのとき, c は I に対して項順序であるという. 2つのコストベクトルが $c_1 \sim c_2$ を満たすとは, ある項順序 $\succ$ に対して $c_1 \sim \succ$ および $c_2 \sim \succ$ が成り立つことをいう.

ここで次の事実が知られている.

Note 3.2. *[11]* $\succ \sim c$ であるための必要十分条件は, I の $\succ$ に関する *Gröbner* 基底 G に対して

$$\mathrm{HT}_{\succ}(g) = \mathrm{in}_c(g) \quad (\forall g \in G)$$

が成り立つことである.

明らかに関係 $\sim$ は同値関係である. $\succ$ に対して

$$C[\succ] := \{c \in \mathbb{R}^n : c \sim \succ\}$$

と定めると, $C[\succ]$ は開凸錐をなすことが知られている.

Definition 3.2. $C[\succ]$ の閉包 $\overline{C[\succ]}$ を I の $\succ$ に関する *Gröbner* 錐と呼ぶ.

Gröbner Walk アルゴリズムは次の事実を利用している.

Note 3.3. *[11]*

$$\mathrm{GF}(I) := \{\overline{C[\succ]}\}_{\succ:\text{項順序}}$$

と定めると, $\mathrm{GF}(I)$ は多面体的扇をなす.

これにより, それぞれの Gröbner 錐に対して唯一の Gröbner 基底を対応させることができる.
Gröbner Walk アルゴリズムでは, 元の項順序 $\succ_1$ と目的とする項順序 $\succ_2$ それぞれに対して,

$$\alpha \cdot c_i > \beta \cdot c_i \;\Rightarrow\; X^{\alpha} \succ_i X^{\beta}$$

を満たすようなコストベクトル c_1 および c_2 が既知で無ければならない. 一般にこのようなコストベクトルを求めることは困難であるが, 辞書式順序や次数付き順序に対しては容易に求めることができる.

Gröbner Walk アルゴリズムの概要を述べる.

Gröbner Walk アルゴリズム

入力: $\succ_1, \succ_2$: 項順序, c_1, c_2 : コストベクトル, $G_1 : \succ_1$ に関する Gröbner 基底.
出力: $G_2 : \succ_2$ に関する Gröbner 基底.

STEP 0: $w_0 := c_1$, $w^* := c_2$, $F_0 := G_1$, $\succ^0 := \succ_1$ とおき, $i = 0$ とする.
STEP 1: $\mathrm{in}_{w_i}(F_i) = \mathrm{HT}_{\succ_2}(F_i)$ であるならば STEP 4 へ. そうでないならば線分 $\overline{w_i w^*}$ 上を w_i から辿り, $\mathrm{in}_{w_i}(F_i) \neq \mathrm{in}_{w_{i+1}}(F_i)$ となるような最も w_i に近い w_{i+1} を求める.

STEP 2: w_{i+1} を $\succ_2$ で細分化した項順序を $\succ^{i+1}$ とし, F_i を **Local conversion procedure** によって I の $\succ^{i+1}$ に関する Gröbner 基底 F_{i+1} に変換する.
STEP 3: $i := i+1$ として STEP 1 へ.
STEP 4: F_i を G_2 として出力.

Local conversion procedure
入力: 項順序 $\succ^i$, $\succ^{i+1}$, $\succ^i$ に関する Gröbner 基底 F_i, コストベクトル w_{i+1}.
出力: $\succ^{i+1}$ に関する Gröbner 基底 F_{i+1}.

STEP 1: $\mathcal{H}_1 := \mathrm{in}_{w_{i+1}}(F)$ とおく.
STEP 2: イデアル $(\mathcal{H}_1)$ $(= \mathrm{in}_{w_{i+1}}(I))$ の $\succ^{i+1}$ に関する Gröbner 基底を計算し, $\mathcal{H}_2$ とおく.
STEP 3: $F_{i+1} := \emptyset$ とおく.
STEP 4: 各 $h \in \mathcal{H}_2$ に対して次を行う.
STEP 4-1: h を $\succ^i$ を用いて $\mathcal{H}_1$ で割ることによって

$$h = \sum_{g \in F_i} p_g \cdot \mathrm{in}_{w_{i+1}}(g)$$

という表示形を得る.
STEP 4-2: 上の結果をもとに,

$$\hat{h} := \sum_{g \in F_i} p_g \cdot g$$

を計算し, $\hat{h}$ を F_{i+1} に加える.
STEP 5: F_{i+1} を出力する.

Local conversion procedure における $\mathcal{H}_1$ は, そのほとんどの元が 1 つの項からなる多項式である. これにより STEP 2 の Gröbner 基底の計算が比較的高速に行える, というのが Gröbner Walk アルゴリズムの利点である. 本論文では特に, トーリックイデアルという 2 項式イデアルを対象としていることにより, STEP 2 はさらに高速に行えることが知られている [9].

このアルゴリズムをもとに b-Groöbner 基底の変換アルゴリズムを構築する. まず, Gröbner 錐のアナロジーとして b-Gröbner 錐というものを定義する.

Definition 3.3. $\succ$ を項順序とし, bG を $\succ$ に関する I_A の *b-Gröbner* 基底とする. また, c をコストベクトルとする. $\succ$ と c が b に関して同値であるとは,

$$\mathrm{in}_c(g) = \mathrm{HT}_{\succ}(g) \ \ (\forall g \in bG)$$

が成り立つことをいう. このとき, $\succ \sim_b c$ と表し, c は I と b に関して項順序であるという. 明らかに $\sim_b$ は同値関係であり, $\sim$ と同様に

$$bC[\succ] := \{c \in \mathbb{R}^n : \succ \sim_b c\}$$

が定義できる. この閉包 $\overline{bC[\succ]}$ を I_A の $\succ$ に関する *b-Gröbner* 錐と呼ぶ.

Gröbner Walk アルゴリズムを b–Gröbner 基底に拡張する際, 次の事実は重要である.

Lemma 3.4. *bG を $\succ$ に関する I_A の b–Gröbner 基底とする. このとき, 任意のコストベクトル $c \in bC[\succ]$ に対して, bG は I_A の c に関する b–Gröbner 基底である.*

証明. b–Gröbner 基底の性質から, $\succ$ に関する I_A の Gröbner 基底 G で,

$$bG = G \cap \bigoplus_{\beta \leq_{\mathcal{M}} b} (I_A)_\beta$$

を満たすものが存在する. これは

$$C[\succ] \subset bC[\succ]$$

を意味する. $C[\succ]$ と $bC[\succ]$ が一致するならば主張は正しい. いま w^* を $C[\succ]$ の境界上にあり, かつ $bC[\succ]$ の境界上にないような点とする. いま $w^* \in \mathbb{R}^n_+$ であるから, $C[\succ]$ と異なる Gröbner 錐 $C[\succ']$ が存在して w^* がその境界上にある [11]. いま, w^* 上で Local conversion procedure を適用し, $\succ$ に関する Gröbner 基底を $\succ'$ に関する Gröbner 基底に変換することを考える. $\mathcal{H}_1$ において 2 項式であるような元については, その A–次数が b よりも $\leq_{\mathcal{M}}$ に関して小さくないことに注意すると, この変換は $G \cap \bigoplus_{\beta \leq_{\mathcal{M}} b}(I_A)_\beta$ に何ら変化をもたらさないことがわかる. □

b–Gröbner Walk アルゴリズムを記述する.

b–Gröbner Walk アルゴリズム

入力: $\succ_1, \succ_2$: 項順序, c_1, c_2 : コストベクトル, G_1 : $\succ_1$ に関する b–Gröbner 基底.
出力: G_2 : $\succ_2$ に関する b–Gröbner 基底.

STEP 0: $w_0 := c_1$, $w^* := c_2$, $F_0 := G_1$, $\succ^0 := \succ_1$ とおき, $i = 0$ とする.
STEP 1: $\mathrm{in}_{w_i}(F_i) = \mathrm{HT}_{\succ_2}(F_i)$ であるならば STEP 4 へ. そうでないならば線分 $\overline{w_i w^*}$ 上を w_i から辿り, $\mathrm{in}_{w_i}(F_i) \neq \mathrm{in}_{w_{i+1}}(F_i)$ となるような最も w_i に近い w_{i+1} を求める.
STEP 2: w_{i+1} を $\succ_2$ で細分化した項順序を $\succ^{i+1}$ とし, F_i を **b–Local conversion procedure** によって I の $\succ^{i+1}$ に関する b–Gröbner 基底 F_{i+1} に変換する.
STEP 3: $i := i + 1$ として STEP 1 へ.
STEP 4: F_i を G_2 として出力.

Local conversion procedure

入力: 項順序 $\succ^i$, $\succ^{i+1}$, $\succ^i$ に関する b–Gröbner 基底 F_i, コストベクトル w_{i+1}.
出力: $\succ^{i+1}$ に関する b–Gröbner 基底 F_{i+1}.

STEP 1: $\mathcal{H}_1 := \mathrm{in}_{w_{i+1}}(F)$ とおく.
STEP 2: イデアル $(\mathcal{H}_1)$ $(= \mathrm{in}_{w_{i+1}}(I) \cap \bigoplus_{\beta \leq_{\mathcal{M}} b}(I_A)_\beta)$ の $\succ^{i+1}$ に関する b–Gröbner 基底を計算し, $\mathcal{H}_2$ とおく.
STEP 3: $F_{i+1} := \emptyset$ とおく.
STEP 4: 各 $h \in \mathcal{H}_2$ に対して次を行う.
STEP 4-1: h を $\succ^i$ を用いて $\mathcal{H}_1$ で割ることによって

$$h = \sum_{g \in F_i} p_g \cdot \mathrm{in}_{w_{i+1}}(g)$$

という表示形を得る.
STEP 4-2: 上の結果をもとに,

$$\hat{h} := \sum_{g \in F_i} p_g \cdot g$$

を計算し, $\hat{h}$ を F_{i+1} に加える.
STEP 5: F_{i+1} を出力する.

これまでの説明により明らかではあるが, 次の定理を示しておく.

Theorem 3.5. *b-Gröbner Walk* アルゴリズムで出力される G_2 は $\succ_2$ に関する *b-Gröbner* 基底である.

証明. アルゴリズムの終了および b-Local conversion procedure の妥当性は, Gröbner Walk アルゴリズムとほぼ同様に示される. なお, STEP 1 で求められる $\mathcal{H}_1$ はイデアル $\mathrm{in}_{w_{i+1}}(I_A)$ の $\succ^i$ に関する b-Gröbner 基底であり, よって

$$(\mathcal{H}_1)\ (= \mathrm{in}_{w_{i+1}}(I) \cap \bigoplus_{\beta \leq_{\mathcal{M}} b} (I_A)_\beta)$$

がいえる. また STEP 4 において, h および $\hat{h}$ の A-次数が $\leq_{\mathcal{M}}$ に関して b より小さいことが Note2.2 からわかる. □

4 結論

本研究では, 基底変換のテクニックを用いた b-Gröbner 基底計算アルゴリズムを提案した. b-Gröbner 基底は [13] で示されている通り, 整数計画問題 $\mathrm{IP}_{A,c}(b)$ の test set に対応する. 様々な整数計画問題, とくに $0-1$ 整数計画問題として定式化された組合せ最適化問題に対してその test set の解析をすることにより, 問題構造をより理解することができると思われるが, Gröbner 基底計算の困難性からこのような研究はほとんど進歩していない. そこで本研究で提案したアルゴリズムが活用できるものと期待できる. また, トーリックイデアルの A-次数による解析もいくつかなされているが [11], 整数計画問題との関連性に主眼をおいた研究も今後の課題である.

なお, アルゴリズムの改良としては, より一般化された整数計画問題:

$$\min\{c \cdot x : Ax = b, x \in \mathbb{N}\},\ \text{ただし},\ A \in \mathbb{Z}^{d \times n},\ b \in \mathbb{Z}^d,\ c \in \mathbb{R}^n.$$

に適合するような工夫などがあげられる.

参考文献

[1] Adams, W.W. and Loustaunau, P. *An introduction to Gröbner Bases*, 2nd edition. Graduate Studies in Mathematics 3, American Mathematical Society, 1996.

[2] Buchberger, B. *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal.* PhD thesis, Math. Inst. University of Innsbruck, Austria,

[3] Collart, S., Kalkbrener, M. and Mall, D. "Converting Bases with the Gröbner Walk". *Journal of Symbolic Computation*, 24, 465–469, 1997.

[4] Conti, P. and Traverso, C. "Gröbner bases and integer programming". Springer Verlag, LNCS 539, *Proceedings AAECC-9 (New Orleans)*, 130–139, 1991.

[5] Cox, D., Little, J. and O'Shea, D. *Ideals, Varieties, and Algorithms.* Undergraduate Texts in Mathematics, Second Edition, Springer Verlag, 1998.

[6] Cox, D., Little, J. and O'Shea, D. *Using Algebraic Geometry.* Graduate Texts in Mathematics 185, Springer Verlag, 1997.

[7] Faugère, J. C., Gianni, P., Lazard, D. and Mora, T. "Efficient computation of zero-dimensional Gröbner bases by change of ordering". *Journal of Symbolic Computation*, 16, 329–344, 1993.

[8] Graver, J. E. "On the foundations of linear and integer linear programming I". *Mathematical Programming*, 9, 207–226, 1975.

[9] Huber, B. and Thomas, R. "Computing Gröbner Fans of Toric Ideals". *Experimental Mathematics*, 9, 321–331, 2000.

[10] Schrijver, A. *Theory of Linear and Integer Programming.* Wiley-interscience series in discrete mathematics, John Willey & Sons, 1986.

[11] Sturmfels, B. *Gröbner Bases and Convex Polytopes.* University Lecture Series 8, American Mathematical Society, 1995.

[12] Thomas, R. "A Geometric Buchberger Algorithm for Integer Programming". *Mathematics of Operations Research*, 20, 864–884, 1995.

[13] Thomas, R. and Weismantel, R. "Truncated Gröbner bases for integer programming". *Applicable Algebra in Engineering, Communication and Computing*, 8, 241–257, 1997.